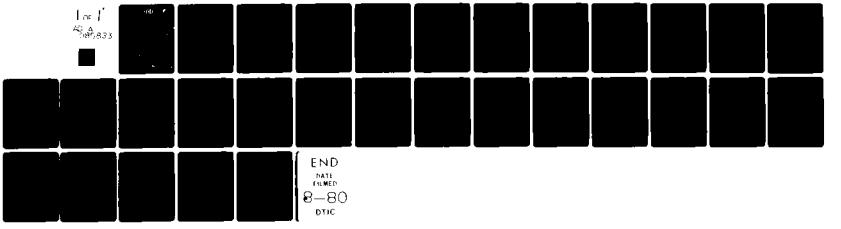
AD-A085 833 MASSACHUSETTS INST OF TECH CAMBRIDGE OPERATIONS RESE--ETC F/G 12/2
THE INTRODUCTION OF FEEDBACK INTO A HIERARCHICAL PRODUCTION PLA--ETC(U)
MAR 80 S C GRAVES N00014-75-C-0556
UNCLASSIFIED TR-177 NL



END
DATE FILMED
8-80
DTIC

LEVEL II

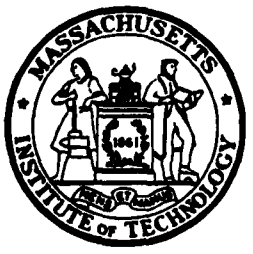
THE INTRODUCTION OF FEEDBACK INTO A HIERARCHICAL PRODUCTION PLANNING SYSTEM

by

STEPHEN C. GRAVES

Technical Report No. 177

OPERATIONS RESEARCH CENTER



MASSACHUSETTS INSTITUTE
OF
TECHNOLOGY

March 1980

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

(9) 1. REPORT NUMBER: Technical Report No. 177

2. GOVT ACCESSION NO.: AD-A085 833

3. RECIPIENT'S CATALOG NUMBER:

(6) 4. TITLE (and Subtitle): THE INTRODUCTION OF FEEDBACK INTO A HIERARCHICAL PRODUCTION PLANNING SYSTEM

5. TYPE OF REPORT & PERIOD COVERED: Technical Report, March 1980

6. PERFORMING ORG. REPORT NUMBER:

(10) 7. AUTHOR(s): Stephen C. Graves

(15) 8. CONTRACT OR GRANT NUMBER(s): N00014-75-C-0556

9. PERFORMING ORGANIZATION NAME AND ADDRESS: M.I.T. Operations Research Center, 77 Massachusetts Avenue, Cambridge, MA 02139 (12) 37

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: NR 347-027

11. CONTROLLING OFFICE NAME AND ADDRESS: O.R. Branch, ONR Navy Dept., 800 North Quincy Street, Arlington, VA 22217 (11)

12. REPORT DATE: Mar[illegible]80

13. NUMBER OF PAGES: 26 pages

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office):

15. SECURITY CLASS. (of this report):

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE:

16. DISTRIBUTION STATEMENT (of this Report)

Releasable without limitation on dissemination.

[illegible] STATEMENT A
Approved for public release;
Distribution Unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Aggregate Production Planning
Lagrangean Relaxation
Hierarchical Production Planning
Production Scheduling

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

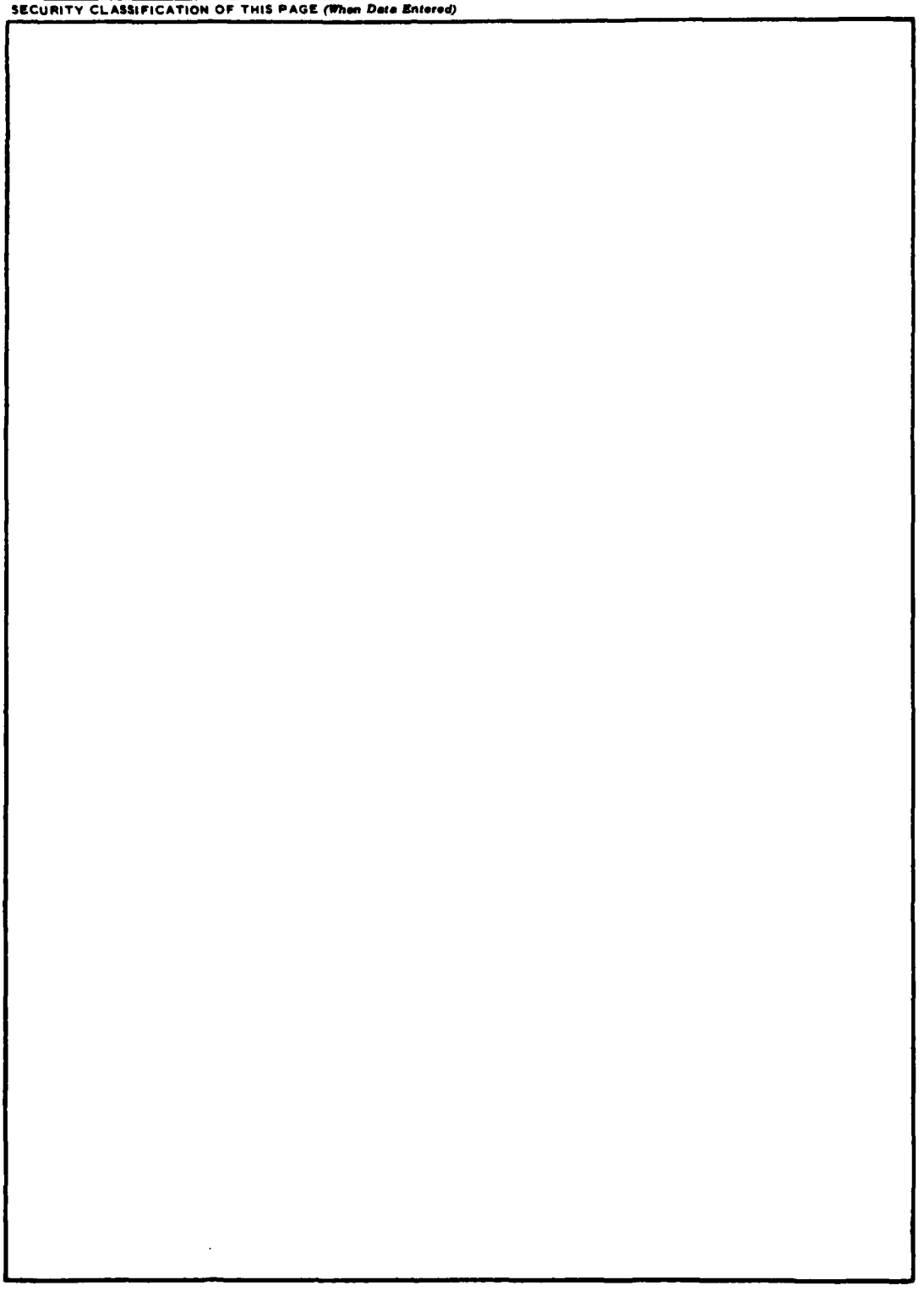
See page ii.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-LF-014-6601

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

270 150



THE INTRODUCTION OF FEEDBACK INTO A HIERARCHICAL PRODUCTION PLANNING SYSTEM

by

STEPHEN C. GRAVES

Technical Report No. 177

Work Performed Under
Contract N00014-75-C-0556, Office of Naval Research
Multilevel Logistics Organization Models
Project No. NR 347-027

Operations Research Center
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139

March 1980

FOREWORD

The Operations Research Center at the Massachusetts Institute of Technology is an interdepartmental activity devoted to graduate education and research in the field of operations research. The work of the Center is supported, in part, by government contracts and grants. The work reported herein was supported by the Office of Naval Research under Contract No. N00014-75-C-0556.

Richard C. Larson
Jeremy F. Shapiro

Co-Directors

ABSTRACT

This paper proposes and tests a framework for decomposing a large scale production planning problem, modeled as a mixed-integer linear program. We interpret this decomposition in the context of Hax and Meal's hierarchical framework for production planning. The procedure decomposes the production planning problem into two subproblems which correspond to the aggregate planning subproblem and a disaggregation subproblem in the Hax-Meal framework. The linking mechanism for these two subproblems is an inventory consistency relationship which is priced out by a set of Lagrange multipliers. The best values for the multipliers are found by an iterative procedure which may be interpreted as a feedback mechanism in the Hax-Meal framework. At each iteration, the procedure finds both a lower bound on the optimal value to the production planning problem and a feasible solution from which an upper bound is obtained. Our computational tests show that the best feasible solution found from this procedure is very close to optimal. For thirty-six test problems the percentage deviation from optimality never exceeds 4.4%, and the average percentage deviation is 2.2%. Twenty-seven of the test problems are mixed-integer linear programs with 240 zero-one variables, while nine test problems have 480 zero-one variables.

1. Introduction

Production planning and scheduling in a batch processing environment are concerned with the efficient utilization and allocation of a firm's production resources so as best to satisfy customer requirements at minimum production costs. Management scientists have devoted a great amount of effort to developing decision models and procedures for this class of problems; Hax [6] gives an insightful survey and discussion of these efforts. Two distinct approaches for production planning and scheduling have been presented in the management science literature. The first approach is a monolithic approach in which the production planning and scheduling problem is formulated and solved as a single mixed-integer programming problem (e.g. Manne [14], Dzielinski and Gomory [4], Lasdon and Terjung [12]). The second approach, given by Hax and Meal [8], is a hierarchical approach for production planning and scheduling. The underlying notion of this approach is to partition the problem into natural subproblems that are consistent with a firm's data processing capabilities, its organizational and responsibility echelons, and its requirements for coordination throughout the organizational structure. In any planning period, the subproblems are solved sequentially, with the solutions of subproblems from the upper hierarchy imposing constraints on the lower hierarchy subproblems. Hax and Golovin [7] provide a current bibliography on hierarchical systems.

The primary advantage of the monolithic approach is that it focuses on a well-defined model formulation for which an optimization is meaningful. In contrast, the hierarchical approach breaks the monolithic problem into subproblems, and at best, can only suboptimize each of these problems. One advantage of the hierarchical approach is that it is computationally simpler

than solving the large mixed-integer programming problem required by the monolithic approach. A second advantage is that the hierarchical approach requires less detailed demand data, in that it needs only aggregate product demand data over the planning horizon, with detailed product demand data over a much shorter scheduling horizon. The monolithic approach requires detailed demand data for the full planning horizon. A third distinction of the hierarchical approach is the extent to which its hierarchical subproblems correspond to the organizational and decision-making levels in the firm; the consequences of this correspondence are the increased interaction between the system and the decision-makers at each level, and the effective coordination of objectives throughout the organization.

The current paper presents a framework for decomposing a monolithic production planning and scheduling problem such that it has a natural interpretation in the context of the hierarchical approach. In this manner the new approach both focuses on a monolithic problem yet still retains the implementation advantages of the hierarchical approach. The new approach may be viewed as providing feedback between two of the subproblems in the hierarchical system. The original framework proposed by Hax and Meal lacks any such feedback mechanism between subproblems. The only interaction is through the constraints generated by higher-echelon subproblems for lower-echelon subproblems. The new approach feeds back information which reflects the cost penalties at the lower-echelon due to the constraints imposed by the higher-echelon subproblems.

The remainder of this paper is organized into four sections. In the next section we formulate the production planning and scheduling problem as a mixed-integer linear program, and review various approaches from the literature for solving this mathematical program. In Section 3 we present a Lagrangean relaxation for the mixed-integer linear program; from this

relaxation both lower bounds and upper bounds on the optimal solution value may be obtained. We also discuss how the Lagrangean relaxation may be viewed as a feedback mechanism in the hierarchical framework proposed by Hax and Meal [8]. Section 4 presents our computational experience with the procedure, while Section 5 provides a discussion of these results and of possible extensions to the procedure.

2. A Production Planning and Scheduling Model

The objective of a production planning and scheduling system is twofold. First, the planning function of the system is to determine what resources are needed at what points in time so as to best satisfy aggregate demand over a specified planning horizon. Second, the scheduling function determines for the immediate scheduling period how the available production resources should be allocated over the individual production items so as to provide the best customer service at the minimum production cost. Typically, the production plan and schedule are generated assuming that all product demand is pre-specified and known over some horizon; then, both the plan and schedule are revised periodically in a rolling-horizon fashion as improved demand forecasts are obtained.

For our study we assume that production items may be aggregated for planning purposes into families, and families aggregated into types. A type is a collection of items that have the same seasonal demand patterns and the same production rate as measured by inventory investment produced per unit time. A family is a set of items within a type, such that the items share a common setup. Such an aggregation scheme was first proposed by Hax and Meal [8], and has been observed in many industrial settings. Hax and Meal [8] provide a good discussion of the advantages of this form of aggregation for production planning.

For purposes of discussion, we will ignore the scheduling of items within a family; this is partially justified by the proposed aggregation scheme in that all production costs can be determined either at the type aggregate or at the family aggregate. We will consider the following simple, single-resource model for scheduling types and families:

$$\text{(PPS)} \qquad \min z = \sum_t (c_t O_t + \sum_i h_{it} I_{it}) + \sum_j \sum_t s_{jt} \delta(P_{jt}) \tag{1}$$

subject to

$$P_{it} + I_{i,t-1} - I_{it} = d_{it} \qquad \forall i,t \tag{2}$$

$$\sum_i k_i P_{it} - O_t \leq r_t \qquad \forall t \tag{3}$$

$$\sum_{j \in T(i)} I_{jt} - I_{it} = 0 \qquad \forall i,t \tag{4}$$

$$P_{jt} + I_{j,t-1} - I_{jt} = d_{jt} \qquad \forall j,t \tag{5}$$

$$O_t, P_{it}, I_{it}, P_{jt}, I_{jt} \geq 0 \qquad \forall i,j,t \tag{6}$$

where subscript i corresponds to types, j to families, and t to time periods. The decision variables for the model are

O_t = overtime worked in period t;

$I_{it}(I_{jt})$ = inventory of type i (family j) in period t;

$P_{it}(P_{jt})$ = production of type i (family j) in period t.

Input data for the model are

c_t = cost premium for overtime in period t;

h_{it} = holding cost for type i, period t;

s_{jt} = setup cost for family j, period t;

$d_{it}(d_{jt})$ = demand for type i (family j) in period t;

r_t = regular production time available in period t;

k_i = production time required per unit of type i;

T(i) = set of families belonging to type i.

$\delta(X)$ is defined to be zero for $X = 0$, and one for $X > 0$.

The planning model (PPS) given in (1) - (6) minimizes overtime costs, holding costs, and setup costs subject to capacity and demand restrictions. As a consequence of the aggregation scheme, production capacity (overtime) costs and inventory holding costs are accounted for by types, while produc-

tion setup costs are accounted for by families. Constraint (3) is the capacity feasibility constraint; constraints (2) and (5) are the inventory balance constraints for types and families, respectively. Constraint (4) links the family inventories to the inventory of the type; this constraint requires that the total inventory for a type equal the sum of the inventories of the families contained in the type. Given that $\sum_{j\epsilon T(i)} d_{jt} = d_{it}$, we can show that constraints (2), (4), and (5) imply that $\sum_{j\epsilon T(i)} P_{jt} = P_{it}$ for all i,t; that is, for each time period total type production equals the sum of the production quantities for its families. This model is possibly the simplest of all planning models in that it considers only one constrained production resource, and it incorporates only a single option, overtime, for varying the resource level. The proposed solution method will be illustrated on this model, but is not limited to such simple models, as will be seen.

The mathematical program given by (1) - (6) is a large-scale mixed-integer linear program which is quite difficult to solve optimally for realistically-sized problems. Manne [14], Dzielinski and Gomory [4], and Lasdon and Terjung [12] have examined a variation of this problem which is also a mixed-integer linear program; the primary difference between these formulations is that [14], [4], and [12] include setup time in the capacity feasibility constraint (3). All three papers solve the problem as a linear program by means of various large scale programming techniques. For instance, the application of Dzielinski and Gomory's approach to (PPS) would consist of dualizing the constraints (2) - (4), and then solving this dual problem by generalized programming [13]. The linear program solution may then be rounded to give a feasible solution; a result of Manne's [14] establishes that the linear program solution will typically be nearly integer.

Hax and Meal [8] discuss the limitations of these approaches, and

present their hierarchical system for attacking (PPS). Their approach is to solve first the aggregate planning subproblem by minimizing overtime and holding costs subject to constraints (2), (3), and (6); this solution specifies the inventory and production levels for each type. The family disaggregation subproblem can then be solved; that is, family setup cost is minimized subject to constraints (4), (5), (6) where the inventory levels for each type in (4) are taken from the aggregate planning model. Hax and Meal solve the aggregate planning subproblem as a linear program, and solve the family disaggregation subproblem for the immediate time period by a heuristic. A critical distinction of the Hax and Meal approach is that it requires detailed family demand estimates only for the immediate time period, and consequently is easier to implement. Bitran and Hax [2] and Bitran, Haas, and Hax [1] give important extensions to this approach. In a similar vein, Jaikumar [11] presents an alternative hierarchical system in which the problem is again decomposed into a planning component and scheduling component. Jaikumar's approach differs from that of Hax and Meal primarily with respect to how the subproblems are linked. Jaikumar uses the solution from the planning subproblem not only to constrain the scheduling subproblem, but also to define certain scheduling costs based on the shadow prices from the planning subproblem.

These hierarchical approaches seem to work well provided that the primary costs are those costs associated with the aggregate planning subproblem; that is, the overtime costs and inventory holding costs associated with the product types are dominant, while the family setup costs are secondary in importance and magnitude. However, when the family setup costs are significant, there is a conceptual gap in the original hierarchical framework in that the aggregate planning subproblem does not consider its impact on total setup costs as determined by the family disaggregation subproblem.

In the next section, we present an iterative procedure to feed back this impact on setup costs to the aggregate planning subproblem.

The approach presented here is similar in spirit to that of Newson [15], who also studied a problem of the type given by (1) - (6). He proposes and tests an iterative heuristic in which the problem is decomposed into a planning subproblem and a scheduling subproblem. Newson's procedure differs from that proposed here primarily with respect to how each subproblem is updated at each iteration.

Bitran, Haas, and Hax [1] have recently proposed a modification to the Hax-Meal framework for settings where the setup costs are significant. This modification entails a clever "look-ahead" procedure for solving the scheduling subproblem; however, the planning subproblem remains unchanged in that it still does not consider its cost impact on the scheduling subproblem.

3. Feedback Procedure

The feedback procedure consists of examining a Lagrangean relaxation (e.g. [5],[16]) to solve a dual problem to (PPS) by an iterative procedure. From this relaxation we immediately obtain a lower bound on the optimal value to (PPS) at each iteration; in addition, we can construct a feasible solution to (PPS) so as to get an upper bound on the optimal solution value. In this manner the procedure can bracket the optimal solution value. Unfortunately, though, there is no theoretical guarantee (either a priori or ex post) for how tightly the optimal solution value may be bracketed by this procedure. To obtain such a (ex post) guarantee would involve incorporating an implicit enumeration scheme into the procedure. Nevertheless, as will be seen, the procedure does quite well and consistently seems to find feasible solutions within 5% of the lower bounds generated from the dual problem.

For any vector of multipliers $\underline{\lambda} = \{\lambda_{it}\}$, a Lagrangean relaxation to (PPS) is obtained by attaching the multipliers to the constraints (4) and bringing these constraints into the objective function:

$$L(\underline{\lambda}) = \min[z + \sum_{i,t} \lambda_{it}(\sum_{j \in T(i)} I_{jt} - I_{it})] \tag{7}$$

$$\begin{cases} \text{s.t.} \quad z = \sum_{t}(c_t O_t + \sum_{i} h_{it} I_{it}) + \sum_{j}\sum_{t} s_{jt}\delta(P_{jt}) & (8) \\ \text{and} \quad (2),\ (3),\ (5),\ (6) \end{cases}$$

The dual problem to (PPS) is now

$$\text{(D)} \qquad \max_{\underline{\lambda}} L(\underline{\lambda}). \tag{9}$$

To solve the dual problem, an iterative procedure is proposed which is consistent with the Hax-Meal hierarchical framework. To see this, note that the Lagrangean relaxation as given in (7), (8) may be separated into the following subproblems:

$$\text{(AP)} \qquad \min z_{AP} = \sum_t [c_t O_t + \sum_i I_{it}(h_{it} - \lambda_{it})]$$

$$\text{s.t.} \quad (2), (3), (6)$$

and

$$\text{(FD)} \qquad \min z_{FD} = \sum_j \sum_t [s_{jt}\delta(P_{jt}) + \lambda_{it} I_{jt}]$$

$$\text{s.t.} \quad (5), (6), \text{ and } i = T^{-1}(j)$$

where $i = T^{-1}(j)$ iff $j \in T(i)$. Clearly (AP) is just an aggregate planning model dealing with the planning of product types, while (FD) is a family disaggregation model dealing with the scheduling of product families. Furthermore (AP) is a linear program, while (FD) may be separated by family into a set of uncapacitated lot-sizing problems, each of which is easily solved by dynamic programming [17]. Hence, the Lagrangean relaxation may be decomposed into two subproblems, each of which is easily solved; furthermore, the two subproblems correspond directly to two linked subcomponents of the Hax-Meal hierarchical framework. It is important to note how the multipliers $\{\lambda_{it}\}$ act to split the holding cost h_{it} between the (AP) model and the (FD) model. We will see that the determination of the appropriate multipliers [or equivalently, the appropriate allocation of the holding cost between the (AP) model and the (FD) model] may be interpretted as a feedback process in the hierarchical framework.

The dual problem is to find $\underline{\lambda}$ to maximize the Lagrangean. A procedure to solve the dual would be iterative, where at each step the Lagrangean is solved for a given $\underline{\lambda}$, and based on this solution a new set of multipliers is picked. A standard framework for this procedure is as follows:

a) set $k = 0$, choose $\underline{\lambda}_o$

b) solve (AP) for $\underline{\lambda} = \underline{\lambda}_k$

c) solve (FD) for $\underline{\lambda} = \underline{\lambda}_k$

d) if the current solution to the Lagrangean satisfies some prespecified "stopping criteria", stop; otherwise set $k = k+1$, update $\underline{\lambda}_k$ and return to b).

Step d) of this procedure is quite vague, with respect to both the "stopping criteria" and the updating of the multipliers; in the next section we illustrate one possible scheme for implementing this procedure. In some sense, the procedure should stop once the solution to (AP) is "consistent" with the solution to (FD); the revision of the multipliers depends upon the current degree of inconsistency between the two subproblems. Nevertheless, we can see the feedback nature of the procedure. For a given value of $\underline{\lambda}$, both the aggregate planning subproblem and the family disaggregation subproblem are solved; based on these solutions the set of multipliers $\underline{\lambda}$ is revised and the first step is repeated. In this manner, the solutions for the individual subproblems are continually revised based on feedback from the other subproblem until a consistent solution is obtained matching the aggregate plan with the family schedules.

The solution of the dual problem given in (9) need not, and usually will not, identify a primal feasible solution to (PPS). In such instances, a duality gap is said to exist and the dual solution provides just a lower bound for the optimal value to (PPS). Two procedures are suggested for resolving these duality gaps. The first approach would be to use the dual problem for generating bounds in a branch and bound or implicit enumeration procedure; the feasibility of such an approach would depend on the tightness of the bounds from the dual problem, and on the number of integer variables (equal to the number of families times the number of time periods). A second possibility is to incorporate the solution of the dual problem into a heuristic procedure. Here, at each iteration in the solution of the

dual, a feasible solution to (PPS), corresponding to the current dual solution, might be generated. The cost for the best of these feasible solutions could be compared with the value of the dual problem, which is a lower bound on the primal problem. The procedure could stop when the best feasible solution was sufficiently close to the lower bound generated from the dual problem. Such a procedure was chosen for closer examination; we report our computational experience in the next section.

A final comment may be made with respect to our expectations for the tightness of the bounds from the dual problem. The Lagrangean relaxation given by (7), (8) [or (AP) and (FD)] does not satisfy the Integrality Property given by Geoffrion [5]. In particular, the optimal value of (FD) does depend on the integrality properties of its variables. As a consequence, we might expect the dual problem to give relatively tight bounds, in that the solution to the dual is at least as good as that given by a linear programming relaxation to (PPS). Furthermore, we should note that the essence of the earlier work by Manne [14], Dzielinski and Comory [4], and Lasdon and Terjung [12] is to solve a linear programming relaxation to (PPS).

4. Computational Study

The computational study consists of solving four sets of problems, each set consisting of nine problems. Details for the test problems are given in the Appendix. For the first three problem sets, the product structure contains twenty families aggregated into three types, where the first two types have five families each and the third type has the remaining ten families. For the fourth problem set there are forty families aggregated into three types with ten families in the first two types and twenty families in the third type. For each problem the planning horizon extends for twelve time periods, the amount of regular time available is constant across these periods, and initial inventories are preset for each family; for convenience the ending inventory for each family after the twelfth period is constrained to be zero.

The first three problem sets differ by the amount of seasonality in the product structure. The first set has no seasonality, the second set has moderate seasonality, while the third set has extreme seasonality. The fourth problem set, which has forty families, has moderate seasonality.

Within each problem set, the vector of family demands {i.e. d_{jt}} is the same for each problem; similarly the overtime cost, and type holding costs are the same for each problem. Within each problem set, the setup cost for each family can assume three values, low, medium, and high; the low cost is one fifth of the medium cost, which is one fifth of the high cost. Within each problem set, the level of available regular time can assume three values, such that total demand over the planning horizon can be covered by 80%, 100%, or 125% of available regular time, respectively. The combination of three possible setup costs with three utilization levels yields nine test problems for each problem set.

The dual problem (9) is solved by a standard subgradient procedure ([3],[9],[10]), in which the multipliers $\{\lambda_{it}\}$ are initialized to be zero. From (7),(8), at a particular value of $\underline{\lambda}$, a subgradient to $L(\underline{\lambda})$ is given by $\underline{\gamma} = \{\gamma_{it}\}$ where

$$\gamma_{it} = \sum_{j \in T(i)} I_{jt} - I_{it} ,$$

and $\{I_{jt}\},\{I_{it}\}$ solves (7),(8). The subgradient procedure updates the current choice for $\underline{\lambda} = \underline{\lambda}_k$ by moving in the direction of the subgradient $\underline{\gamma}_k$:

$$\underline{\lambda}_{k+1} = \underline{\lambda}_k + \alpha_k \underline{\gamma}_k \qquad (10)$$

The scalar α_k is the step size, and its choice is critical to the behavior of the procedure [3].

For our computational tests, we had the best experience on the first three problem sets using step sizes generated from the following sequence: $\alpha_k = 1$ for k=1,2,...,10; $\alpha_k = .5$ for k=11,...,20; $\alpha_k = .25$ for k=21,...,30;... For the fourth problem set the sequence chosen was $\alpha_k = 1$ for k=1,...,20; $\alpha_k = .5$ for k=21,...,30; $\alpha_k = .25$ for k=31,...,40;... For each problem we terminated the subgradient procedure after 60 iterations.

At each iteration a feasible solution to (PPS) is generated from the dual solution based on the family schedules from the solution to (FD). Note that the solution from the (FD) problem satisfies all demand constraints and is always feasible in (PPS) since we assume there is unlimited overtime available in each period; if this were not so, a slightly more complicated procedure would be needed for finding feasible solutions. By evaluating the cost of this solution in (PPS) we obtain an upper bound on (PPS). We then try to improve this feasible solution by myopically smoothing this production plan. For this heuristic, we consider the periods of the planning horizon in reverse order, skipping over those periods in which no overtime is

planned in the current production plan. Having found a period in which production exceeds available regular production time, we try to reschedule production in this period to earlier periods with unused regular production time. For instance suppose overtime is required in period t" and consider family j such that $P_{jt"} > 0$; then we need identify the latest period t' such that $t' < t"$, $P_{jt'} > 0$ and unused regular production time is available. We will reschedule as much of $P_{jt"}$ to time period t' as is economic in that the additional holding costs over the interval [t',t") are less than the overtime costs saved in period t"; if all of $P_{jt"}$ may be rescheduled, then we also save a setup cost for family j. In this manner, we obtain an improved feasible solution.

Tables 1-3 report the percentage discrepancy in cost between the best feasible solutions to (PPS) and the lower bound from the dual problem to (PPS) for each of the nine problems for the first three problem sets, respectively. Note that for these cases (PPS) is a mixed-integer linear program with 240 zero-one variables. It is encouraging to report from these tables that these percentages, which represent the maximum deviation from optimality of the best known solution, are remarkably and consistently small; furthermore, the performance of the procedure seems to be relatively insensitive to the magnitude of the setup costs, to the level of resource utilization, and to the amount of demand seasonality. Table 4 reports the performance of the procedure for the fourth problem set which contains forty families to schedule. Here (PPS) is a mixed-integer linear program with 480 zero-one variables. The results in Table 4 are comparable to those in Tables 1 - 3, giving evidence that the performance of the procedure is maintained for larger problems. In addition, Table 4 reports the results from running the subgradient procedure beyond 60 iterations to 100 iterations; we see that the additional improvement from the longer run is quite small.

Table 1: Percentage difference between best feasible solution and lower bound for problem set 1 (no seasonality)

Setup Cost	Resource Utilization 80%	100%	125%
low	0.2%	1.3%	1.4%
medium	0.7%	2.9%	1.2%
high	2.8%	4.0%	2.7%

Table 2: Percentage difference between best feasible solution and lower bound for problem set 2 (moderate seasonality)

Setup Cost	Resource Utilization 80%	100%	125%
low	1.4%	2.1%	1.4%
medium	2.1%	2.6%	1.9%
high	2.7%	2.4%	1.4%

Table 3: Percentage difference between best feasible solution and lower bound for problem set 3 (extreme seasonality)

Setup Cost	Resource Utilization 80%	100%	125%
low	3.1%	2.4%	0.2%
medium	4.4%	2.8%	1.4%
high	3.1%	4.4%	3.8%

Table 4: Percentage difference between best feasible solution and lower bound for problem set 4 (forty families)*

Setup Cost	Resource Utilization 80%	100%	125%
low	1.0% (1.0%)	2.9% (2.4%)	1.8% (1.8%)
medium	1.2% (1.1%)	3.6% (3.5%)	1.5% (1.5%)
high	2.4% (2.3%)	1.5% (1.4%)	1.0% (1.0%)

*The percentages in parentheses are the results from stopping the subgradient procedure after 100 iterations.

The computational experiments were conducted on a PRIME 400 minicomputer.* For the twenty-seven problems from sets 1-3, the computation time per problem ranged from 90 cpu seconds to 364 cpu seconds, with the average time per problem being 236 cpu seconds. The observed wide variation in times is due to the variation across problems for solving the linear program (AP). At each iteration of the procedure the objective function of (AP) is modified and the linear program must be resolved; depending upon the extent of the modification, the new linear program may or may not be easily solved. For the nine problems in set 4, the average cpu time was 311 seconds with a range of 147 to 405 seconds. Hence, doubling the problem size in terms of number of families seems to increase the cpu time by only one third. Note, however, that although the number of families were doubled, the number of types remained at three; hence the size of the linear program for (AP) was unchanged. This suggests that the set up and solution of the linear programs account for much of the reported computational time. We solve the linear programs using the SEXOP (Subroutines for Experimental Optimization) code, which is an experimental code for algorithmic research written by Professor Roy Marsten, University of Arizona. We might expect quicker solution times with a commercial linear programming code.

* The PRIME 400 minicomputer is three to eight times slower than an IBM 370/168, depending upon the type of operation.

5. Discussion and Extensions

This paper proposes a framework for decomposing a large scale production planning problem. This decomposition is quite natural in that the resulting subproblems correspond to components in Hax-Meal's hierarchical framework for production planning. The underlying basis for the decomposition is to separate the capacity planning portion of the problem from the lot-sizing problem; the linking mechanism for these two components is an inventory consistency relationship [i.e. constraint set (4)] which may be priced out by a set of multipliers. In this fashion, a Lagrangean function is defined which may be optimized by a subgradient procedure to obtain a lower bound for the production planning problem. A simple heuristic is proposed for converting each Lagrangean solution into a feasible solution for the production planning problem so as to get a corresponding upper bound.

The computational results presented indicate that the proposed framework may be quite effective for generating good production schedules for quite difficult combinatorial problems. Indeed for the thirty-six test problems the best solutions were always within 4.4% of a lower bound to the optimum, and were within 3.1% of the lower bound to the optimum in all but five cases. Furthermore these results seem reasonably robust with respect to the problem specification and problem size.

We should note that although the proposed framework has been illustrated with respect to (PPS) given by (1) - (6), it is not limited to this formulation of the production planning problem. The proposed framework is equally applicable to extensions of (1) - (6) in which there may be more than one type of constrained capacity and/or more than one option for adjusting capacity levels such as changing the work force level. In these instances the linear program for (AP) will be slightly more complex to handle the additional complications.

Extensions of the current study might concentrate on various modeling considerations. For one, as noted by Hax and Meal [8], it may be unrealistic to expect to be able to generate meaningful demand forecasts for the families for the later periods in the planning horizon. Hence, we may desire to schedule families over a shorter horizon with more detailed periods than those used for types. For instance, families might be scheduled by week for the next twelve weeks (three months), while the production of types would be planned by month over the next twelve months; for this example the formulation in (1) - (6) could be modified by stating the consistency constraints (4) only for the ending inventories after the first three months (weeks 4, 8, and 12). The computational procedure would not be affected by this modification, with one exception. The family disaggregation subproblem (FD) would have to be modified to force the inventory at the end of its horizon (e.g. week 12) to correspond to that planned in the aggregate planning subproblem (AP).

Another modeling consideration is with respect to how types and families are defined; that is, what is the best way to aggregate items into families, and families into types. We hope that the proposed framework might provide insight for examining the implications of various aggregation strategies.

Finally, it would be important to consider how such a production planning system would be implemented in practice. The planning problem given by (1) - (6) is posed as if all demand is known with certainty and all schedules are to be frozen over the planning horizon. However, most production planning seems to be done on a periodic (rolling horizon) basis in which the demand data is continually being revised as the demand forecasts become less uncertain. Implicit in our research and that of many other researchers, has been the assumption that the production schedules that are generated

periodically from static planning models such as (1) - (6), are effective when implemented in a dynamic setting. We need further work to examine the validity of this assumption.

Acknowledgement

The author wishes to acknowledge the very helpful comments of Professor Thomas L. Magnanti on an earlier draft of this paper.

References

1. Bitran, G. R., E. A. Haas, and A. C. Hax, "Hierarchical Production Planning. Part I - A Single Stage System", Working paper, February 1980.

2. Bitran, G. R., and A. C. Hax, "On the Design of Hierarchical Production Planning Systems", Decision Sciences, Vol. 8, No. 1, January 1977.

3. Crowder, H., "Computational Improvements for Subgradient Optimization", in Symposia Mathematica, Vol. 19, Academic Press, 1976, pp. 357-372.

4. Dzielinski, B. P., and R. E. Gomory, "Optimal Programming of Lot Sizes, Inventory and Labor Allocations", Management Science, Vol. 11, No. 9, July 1965, pp. 874-890.

5. Geoffrion, A. M., "Lagrangean Relaxation for Integer Programming", Math. Programming Study 2, pp. 82-114.

6. Hax, A. C., "Aggregate Production Planning", in Handbook of Operations Research, J. Moders and S. Elmaghraby (editors), Van Nostrand Reinhold, 1978.

7. Hax, A. C., and J. Golovin, "Hierarchical Production Planning Systems", in Studies in Operations Management, A. C. Hax (editor), Amsterdam, North Holland, 1978.

8. Hax, A. C., and H. C. Meal, "Hierarchical Integration of Production Planning and Scheduling", in North Holland/TIMS, Studies in Management Sciences, Vol. 1, Logistics, North Holland-American Elsevier, 1975, pp. 53-69.

9. Held, M., and R. M. Karp, "The Traveling Salesman Problem and Minimum Spanning Trees: Part II", Math. Programming, Vol. 1, pp. 6-25.

10. Held, M., P. Wolfe, and H. P. Crowder, "Validation of Subgradient Optimization", Mathematical Programming, Vol. 6, 1974, pp. 62-88.

11. Jaikumar, R., "An Operational Optimization Procedure for Production Scheduling", Computers and Operations Research, Vol. 1, No. 2, August 1974, pp. 191-200.

12. Lasdon, L. S., and R. C. Terjung, "An Efficient Algorithm for Multi-Item Scheduling", Operations Research, Vol. 19, No. 4, July-August 1971, pp. 946-969.

13. Magnanti, T. L., J. F. Shapiro, and M. H. Wagner, "Generalized Linear Programming Solves the Dual", Management Science, Vol. 22, No. 11, July 1976, pp. 1195-1203.

14. Manne, A. S., "Programming of Economic Lot Sizes", Management Science, Vol. 4, No. 2, January 1958, pp. 115-135.

15. Newson, E. F. P., "Multi-Item Lot Size Scheduling by Heuristic. Part II: With Variable Resources", Management Science, Vol. 21, No. 10, June 1975, pp. 1194-1203.

16. Shapiro, J. F., "A Survey of Lagrangean Techniques for Discrete Optimization", in Annals of Discrete Mathematics 5: Discrete Optimization, P. L. Hammer, E. L. Johnson, and B. H. Kortz (editors), North Holland, pp. 113-138.

17. Wagner, H. M., and T. Whitin, "Dynamic Version of the Economic Lot Size Model", Management Science, Vol. 5, No. 1, October 1958, pp. 89-96.

Appendix

Specification of Test Problems for Computational Experiments

For the computational experiments we generated four problem sets, each with nine test problems. The specifications for the generation of these test problems are given below.

Product Structure: For each problem set, there are three product types; the assignment of families across these types is given below:

	number of families in type 1	type 2	type 3	total families
problem set 1-3	5	5	10	20
problem set 4	10	10	20	40

Overtime Cost: The overtime cost c_t is 5.00 for all time periods for all test problems.

Holding Costs: The inventory holding cost h_{it} is 1.00 for type 1, 1.75 for type 2, and 1.50 for type 3 for all time periods for all test problems.

Setup Costs: For each type a range $(\underline{S}_i, \overline{S}_i)$ is specified as given below:

	Type 1	Type 2	Type 3
$\underline{S}_i$	50.0	100.0	100.0
$\overline{S}_i$	150.0	200.0	200.0

For each family j in type i we set s_j by a random drawing from a uniform distribution over the range $(\underline{S}_i, \overline{S}_i)$; s_j is the "medium" setup cost for family j for all time periods for all test problems. To get the "low" setup cost we divide s_j by five, while to get the "high" setup cost we multiple s_j by five.

Family Demand: For problem set k, demand for family j in period t is given by

$$d_{jt}^{(k)} = f_{it}^{(k)} * u_j$$

where $j \in T(i)$, $f_{it}^{(k)}$ is the multiplicative seasonality factor for type i in period t in problem set k, and u_j is a random draw from a uniform distribution over the range $(\underline{D}_j, \overline{D}_j)$. The seasonality factors $f_{it}^{(k)}$ are given below:

		Time Period											
		1	2	3	4	5	6	7	8	9	10	11	12
problem set 1	type 1	1.0	1.0	→	→	→	→	→	→	→	→	→	1.0
	type 2	1.0	1.0	→	→	→	→	→	→	→	→	→	1.0
	type 3	1.0	1.0	→	→	→	→	→	→	→	→	→	1.0
problem set 2&4	type 1	1.0	1.0	→	→	→	→	→	→	→	→	→	1.0
	type 2	0.8	0.8	0.7	0.5	0.7	1.0	1.0	1.2	1.3	1.5	1.2	1.0
	type 3	1.0	1.0	1.0	1.2	1.3	1.5	1.3	1.0	0.9	0.7	0.6	0.8
problem set 3	type 1	1.0	0.6	0.0	0.0	0.0	0.0	0.0	0.8	1.6	3.0	3.0	2.0
	type 2	0.8	0.6	0.3	0.0	0.0	0.6	1.2	1.5	2.0	2.0	1.5	1.2
	type 3	0.3	0.5	0.6	1.0	1.2	1.5	2.0	2.2	1.3	1.0	0.5	0.2

The demand ranges for families 1-20 are given on the following page.
The demand ranges for families 21-40 for problem set 4 are the same as that for families 1-20 respectively. The type demand d_{it} is obtained by aggregating family demand.

Initial Inventory: For each family j we compute the economic order quantity Q_j based on its average demand rate, holding cost and setup cost. The family's initial inventory is found by a random draw taken from a uniform distribution over the range $(0, Q_j)$. The initial inventory is then credited towards the family demand to obtain a net family demand. Type demand is

Families

	type 1					type 2					type 3									
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
$\underline{D}_j$	20	40	15	25	80	80	15	40	40	55	20	20	30	30	20	30	30	50	60	40
$\overline{D}_j$	40	60	25	65	120	120	25	60	60	85	40	40	50	50	40	70	50	100	90	80

redefined accordingly.

Available Regular Time: For problem set k, we compute total aggregate production requirements $R^{(k)}$ from the type demands; then the available regular time in period t for problem set k is given by

$$R_t^{(k)} = R^{(k)}/12$$

where 100% resource utilization is assumed. For the case of 80% resource utilization we multiply $R_t^{(k)}$ by 1.20, while for 125% resource utilization we multiply $R_t^{(k)}$ by 0.80.